

**Yee &  
Associates, P.C.**

4100 Alpha Road  
Suite 1100  
Dallas, Texas 75244

Main No. (972) 385-8777  
Facsimile (972) 385-7766

## Facsimile Cover Sheet

**RECEIVED  
CENTRAL FAX CENTER**

**MAY 23 2005**

<b>To:</b> Commissioner for Patents for Examiner Ellen C. Tran Group Art Unit 2134	<b>Facsimile No.:</b> (703) 872-9306
<b>From:</b> Carrier Parker Legal Assistant to Ted Fay	<b>No. of Pages Including Cover Sheet:</b> 30
<b>Message:</b>  Enclosed herewith: <ul style="list-style-type: none"><li>• Transmittal Document; and</li><li>• Appeal Brief.</li></ul>	
<b>Re:</b> Application No. 09/755,564 Attorney Docket No.: FR20000023US1	
<b>DATE:</b> Monday, May 23, 2005	
<b>Please contact us at (972) 385-8777 if you do not receive all pages indicated above or experience any difficulty in receiving this facsimile.</b>	<i>This Facsimile is intended only for the use of the addressee and, if the addressee is a client or their agent, contains privileged and confidential information. If you are not the intended recipient of this facsimile, you have received this facsimile inadvertently and in error. Any review, dissemination, distribution, or copying is strictly prohibited. If you received this facsimile in error, please notify us by telephone and return the facsimile to us immediately.</i>

**MAY 24 2005  
OPE/JCWS**

**RECEIVED**

**PLEASE CONFIRM RECEIPT OF THIS TRANSMISSION BY  
FAXING A CONFIRMATION TO 972-385-7766.**

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Lamberton et al.**Serial No.: **09/755,564**Filed: **January 5, 2001**For: **Methods and Systems for  
Defeating TCP SYN Flooding Attacks****36736**PATENT TRADEMARK OFFICE  
CUSTOMER NUMBER§  
§  
§  
§  
§  
§Group Art Unit: **2134**Examiner: **Tran, Ellen C.**Attorney Docket No.: **FR20000023US1**Certificate of Transmission Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (703) 872-9306 on May 23, 2005.

By: Carrie Parker

Carrie Parker

TRANSMITTAL DOCUMENT

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

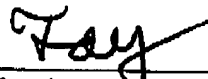
Sir:

ENCLOSED HEREWITH:

- Appeal Brief (37 C.F.R. 41.37).

A fee of \$500.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0461. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0461. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0461.

Respectfully submitted,



Theodore D. Pay, III

Registration No. 48,504

Duke W. Yee

Registration No. 34,285

YEE &amp; ASSOCIATES, P.C.

P.O. Box 802333

Dallas, Texas 75380

(972) 385-8777

ATTORNEYS FOR APPLICANTS

Docket No. FR20000023US1

PATENT

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Lamberton et al.

Serial No. 09/755,564

Filed: January 5, 2001

For: Methods and System for Defeating  
TCP SYN Flooding Attacks§  
§  
§  
§  
§  
§  
§

Group Art Unit: 2134

Examiner: Tran, Ellen C.

RECEIVED  
CENTRAL FAX CENTER

MAY 23 2005

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450Certificate of Transmission Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being transmitted via  
facsimile to the Commissioner for Patents, P.O. Box 1450,  
Alexandria, VA 22313-1450, facsimile number (703) 872-9306  
on May 23, 2005.

By:

Carrie Parker

Carrie Parker

## APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on March 23, 2005.

The fees required under § 41.20(B)(2), and any required petition for extension of time for filing this  
brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

05/24/2005 KBETEM1 00000049 090461 09755564

01 FC:1402 500.00 DA

(Appeal Brief Page 1 of 28)  
Lamberton et al. - 09/755,564

**REAL PARTY IN INTEREST**

The real party in interest in this appeal is the following party: International Business Machines Corporation.

**RELATED APPEALS AND INTERFERENCES**

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

**STATUS OF CLAIMS**

**A. TOTAL NUMBER OF CLAIMS IN APPLICATION**

Claims in the application are: 1-5 and 7-14

**B. STATUS OF ALL THE CLAIMS IN APPLICATION**

1. Claims canceled: 6
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1-5 and 7-14
4. Claims allowed: None
5. Claims rejected: 1-5 and 7-14
6. Claims objected to: None

**C. CLAIMS ON APPEAL**

The claims on appeal are: 1-5 and 7-14

**STATUS OF AMENDMENTS**

No amendments were filed after the first final office action of January 25, 2005.

**SUMMARY OF CLAIMED SUBJECT MATTER****A. CLAIM 1 - INDEPENDENT**

The following explanation of the subject matter of claim 1 does not limit or otherwise modify claim 1 as presented in the appendix of claims. The subject matter of claim 1 is directed to a method of defeating a SYN flooding attack in a server computer. A SYN flooding attack is a type of denial of service attack. A denial of service attack attempts to overwhelm a server with a vast number of spurious communications. The SYN flooding attack may be used with a distributed denial of service attack, in which a malicious user can take over a vast number of computers to direct an automated attack against any computer connected to the Internet.

The SYN flooding attack, in particular, uses the normal SYN-ACK process that allows a client to establish a transmission control protocol (TCP) connection to a server. During the normal process, the client sends a SYN (synchronizing sequence number) message to a server. The server then acknowledges the SYN message by transmitting a SYN-ACK message to the client. The client then finishes establishing the connection by responding with an ACK (acknowledgement) message to the server. The connection between the client and the server is then open such that service-specific data may be exchanged between the client and the server. During a SYN flooding attack, thousands or even millions of SYN messages are transmitted to a server from many different client computers. The server, which has a limited capability for handling SYN messages, becomes overwhelmed. The server cannot process legitimate traffic or processes legitimate traffic slowly. The server may also shut down or freeze as a result of the attack.

The method of claim 1 solves this problem by, among the other claimed steps, computing an initial sequence number receiver side (ISR), embedding the ISR in the SYN-ACK message, and responsive to receiving an ACK message, determining whether to establish a transmission control block for the client by evaluating an incremented value of the ISR included in the ACK message. Thus, the denial of service attack is much more likely to be defeated.



**B. CLAIM 7 - INDEPENDENT**

The subject matter of claim 7 is similar to the subject matter of claim 1. Claim 7 includes, among the other claimed steps, the step of responsive to evaluating the value of the initial sequence number receiver side as an authentic computed initial sequence number receiver side, allocating resources for a transmission control protocol connection according to content specified in a previously received SYN message.

**C. CLAIM 11 - INDEPENDENT**

The subject matter of claim 11 is similar to the subject matter of claim 1. Claim 11 is directed to a computer program product for defeating a SYN flooding attack.

**D. CLAIM 14 - INDEPENDENT**

The subject matter of claim 14 is similar to the subject matter of claim 1. Claim 14 is directed to a system for defeating a SYN flooding attack.

**GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

**A. GROUND OF REJECTION 1 (Claims 1-5 and 7-14)**

Claims 1-5 and 7-14 stand rejected under 35 U.S.C. § 102(e) as anticipated by *Denker*, Communication Protocol with Improved Security, U.S. Patent 5,958,053 (Sep. 28, 1999).

### ARGUMENT

#### A. GROUND OF REJECTION 1 (Claims 1-5 and 7-14)

##### A.1. Claim 1

The examiner rejects claim 1 as anticipated by *Denker, Communication Protocol with Improved Security*, U.S. Patent 5,958,053 (Sep. 28, 1999). The examiner states that:

As to independent claim 1, "A method for defeating, in a server unit of an Internet Protocol network, a SYN flooding attack, said server unit running Transport Control Protocol to allow the establishment of one or more transmission control protocol connections with one or more client units, said method comprising the steps of: upon having activated the transmission control protocol in said server unit:" is taught in '053 col. 4, lines 44-55;

"listening for the receipt of a SYN message sent from a client unit" and "resuming to said listening step" is shown in col. 6, lines 59-60;

"upon receiving said SYN message: computing an Initial Sequence number Receiver side; wherein said Initial Sequence number Receiver side is embedded with connection parameters specified in the SYN message; responding to said client unit with a SYN-ACK message including said computed said Initial Sequence number Receiver side;" is disclosed in col. 4, line 58 through col. 5, line 43;

"responsive to receiving an ACK message, determining whether to establish a transmission control block for the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message" is shown in '053 col. 5, lines 37-43.

Office Action of January 25, 2005, p. 3-4 (emphasis in original). In the response to arguments section of the final office action, the examiner states that:

In response to applicants argument on page 10 "Thus, *Denker* ('053) fails to describe or suggest a mechanism for embedding an initial sequence number receiver side "with connection parameters specified in the SYN message". The Office disagrees. '053 shows embedding an initial sequence number in col. 4, line 63 through col. 5, line 43 "This ACK message (in addition to the information required by standard TCP) includes the encoded value and repeats the clients requested options. A counter associated with each address in the Friends Table can be used to keep track of the number of successful connections established."

In response to applicants' argument on page 10, the reference does not

describe “determining whether to establish a transmission control block for the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message.”

The Office disagrees. ‘053 show evaluating the incremented value in col. 5, lines 37-13 “A counter associated with each address in the Friends Table can be used to keep track of the number of successful connections established as compared to the total number of connection requests from the client, and allow the server to expunge the client’s address from the Friends Table if there are too many unsuccessful connection attempts.”

Office Action of January 25, 2005, p. 2-3 (emphasis in original).

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983).

Claim 1 is presently as follows:

1. A method for defeating, in a server unit of an Internet Protocol network, a SYN flooding attack, said server unit running Transmission Control Protocol to allow the establishment of one or more transmission control protocol connections with one or more client units, said method comprising the steps of:
  - upon having activated the transmission control protocol in said server unit,
  - listening for the receipt of a SYN message sent from a client unit;
  - upon receiving said SYN message,
  - computing an Initial Sequence number Receiver side, wherein said Initial Sequence number Receiver side is embedded with connection parameters specified in the SYN message;
  - responding to said client unit with a SYN-ACK message including said Initial Sequence number Receiver side;
  - resuming to said listening step; and
  - responsive to receiving an ACK message, determining whether to establish a transmission control block for the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message.

*Denker* does not anticipate claim 1 because *Denker* does not show or suggest the claimed step of: responsive to receiving an ACK message, determining whether to establish a

transmission control block for the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message.

The examiner states otherwise in both the initial rejection and in the response to arguments, citing from *Denker* as follows:

According to TCP2E, if the client's address is on the server's Friends Table, the connection request (i.e., received SYN message) is processed according to TCP. *A counter associated with each address in the Friends Table can be used to keep track of the number of successful connections established as compared to the total number of connection requests from the client, and allow the server to expunge the client's address from the Friends Table if there are too many unsuccessful connection attempts.*

*Denker*, col. 5, l. 34-43 (emphasis shows portion cited by the examiner).

However, the examiner's characterization of the cited portion of *Denker* is incorrect. The cited portion of *Denker* states that a counter is associated with each address in a friends table. The friends table is used to track the number of *successful connection requests* from the client. In contrast, the claimed step requires determining whether to establish a transmission control block in the first place by evaluating *an incremented value* of the Initial Sequence number Receiver side *included in the ACK message*.

The cited portion in *Denker* does not show or suggest a step of determining whether to establish a transmission control block for the client by evaluating anything *included in the ACK message*, as claimed. Instead, the cited portion of *Denker* teaches a friends table that tracks addresses of clients *for which a connection has already been made*. Any number being incremented by *Denker* is not associated with the ACK message as claimed, because the ACK message is required to establish the connection in the first place. In other words, the friends table taught in *Denker* has nothing to do with determining whether to establish a transmission control block, as claimed. Thus, the cited portion of *Denker* does not show or suggest determining whether to establish a transmission control block by evaluating an incremented value of the Initial Sequence number Receiver side, as claimed.

In addition, the cited portion of *Denker* does not show or suggest determining whether to establish a transmission control block for a client by evaluating an *incremented value* of the Initial Sequence number Receiver side. Instead, *Denker* determines whether to establish a

transmission control block by performing an unspecified mathematical function on *both* the client address and a "secret," which may be a random number, as shown by the following passages from *Denker*:

In the TCP2B protocol according to an embodiment of the present invention, the client requests a TCP connection with a server using a SYN message. The client indicates its support for the TCP2B protocol of the present invention using one or more bits of the TCP header (such as the OPT field). In response to receiving the SYN message, the server then sends a SYNACK message indicating the server's support for the TCP2B protocol. *The SYNACK message includes an encoded value as a mathematical (i.e., cryptologic) function of at least the client's address and a secret known only to the server.* In response to the SYNACK message indicating the server's support for TCP2B, the client sends an ACK message to the server. This ACK message (in addition to the information required by standard TCP) includes the encoded value and repeats the client's requested options. The server then analyzes the encoded value in the ACK message to determine if it passes the appropriate mathematical (i.e., cryptologic) test. *If the encoded value included in the ACK message passes the appropriate mathematical test, then the client is properly complying with the TCP2B protocol, and the server allocates a full Transmission Control Block in memory, and the connection becomes fully established.*

*Denker*, col. 4, l. 53 through col. 5, l. 8 (emphasis supplied).

If the client's address is not on the server's Friends Table, the server calculates an encoded value. *The encoded value is calculated as the mathematical function of at least the client's address and a secret (i.e., a random number) known only to the server.* The server sends an ACK message to the client including the calculated encoded value as the acknowledgment number. Because the acknowledgment number does not acknowledge any messages previously sent by the client, this ACK message appears to the client as a half-open connection. The client responds by sending a reset message to the server, as required by the standard TCP specifications. If the reset message passes a mathematical test at the server, the server adds the client's address to the Friends Table. Then, in accordance with standard TCP, the client will again attempt to establish a TCP connection with the server by re-issuing the SYN message to the server. This SYN message (or packet) will be welcomed by the server, since the client's address is now in the Friends Table.

*Denker*, col. 5, ll. 16-33 (emphasis supplied).

*Denker* plainly determines, responsive to an ACK message, whether to establish a transmission control block by performing an unspecified mathematical function on both the client address and a random number. *Denker* does not show or suggest the claimed step of

determining whether to establish a transmission control block by evaluating an incremented value of the Initial Sequence number Receiver side, as claimed. Furthermore, *Denker* does not show or suggest that the random number is incremented by one. In the light that an unspecified mathematical function is performed on *both* the client address and a random number, one of ordinary skill would have no reason to assume that the mathematical function only increments the random number by one. Even if one of ordinary skill could make the assumption, the question of anticipation is resolved by whether the claim reads on the process that *Denker* discloses, not on what *Denker* broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Thus, even if *Denker* broadly taught the claimed step, that fact would be insufficient to establish that *Denker* anticipates claim 1. In addition, because one of ordinary skill would have no reason to believe that the cited portion of *Denker* could teach the claimed invention, claim 1 is also non-obvious.

Claims 2, 11, 12, and 14 stand or fall with claim 1. Claims 11 and 14 are independent claims containing limitations similar to those present in claim 1.

#### A.2. Claim 3

The examiner rejects claim 3 as anticipated by *Denker*, further stating that:

As to dependent claim 3, "wherein said computing step further comprises the steps of: updating, in a server unit, a pseudo-random number {PRN} generator; holding a current key; remembering a former key; and using said current key as said randomly generated key for said computed Initial Sequence number Receive side" is shown in '053 col. 10, line 50 through col. 11, line 19.

Office Action of January 25, 2005, p. 4 (emphasis in original).

Claim 3 is as follows:

3. The method according to claim 2, wherein said computing step further comprises the steps of:
  - updating, in said server unit, a pseudo-random number (PRN) generator;
  - holding a current key;
  - remembering a former key; and
  - using said current key as said randomly generated key for said Initial Sequence number Receiver side.

*Denker* does not anticipate claim 3 because *Denker* does not show the steps of updating a

PRN, holding a current key, remembering a current key, and using a current key as claimed. The examiner asserts otherwise, citing the following portions of *Denker*:

In accordance with TCP, client 105 then sends a SYN message at step 4050D in a second attempt to establish the new connection with server 110. Naturally, the SYN message of step 4050D includes a restatement of the client-specified options. Steps 4050D, 5060D and 6070D of FIG. 5 are the same as the three step handshake (steps 1020A, 2030A and 3040A, respectively) of FIG. 1 to establish a TCP connection. After step 6070D, client 105 and server 110 have recovered from the half-open connection and fully reestablished the new connection.

#### TCP2E Protocol

According to an embodiment of the present invention, TCP2E relies upon the ability of a standard TCP client 105 to handle a half-open connection, and uses this to provide the server 110 with an improved defense to a SYN Flood attack.

Prior to step 1020E of the TCP2E protocol (FIG. 6), server 110 previously stored general information, including its own IP address, and a secret known only to the server 110. The secret is typically used for a plurality of connection requests.

According to an embodiment of the present invention, server 110 stores in memory a Friends Table, which is a table (such as a hash-table or a list) maintained in server 110's memory containing the IP addresses (or other identifying information) of clients that have been recently observed to be complying with important parts of the TCP protocol. The Friends Table may be of constant size, such as one thousand IP addresses. In one embodiment of the present invention, server 110 adds a client's IP address to the server's Friends Table after a TCP connection has been established between the server and client using any first level protocol (i.e., TCP, Bernstein/Schenk Syncookie method, TCP2B, TCP2E). Also, importantly, a client's address can be added to the server's Friends Table when the client has complied with the initial steps of TCP2E as described below. If it is necessary to add a new IP address to the Friends Table and there are no free slots, a slot can be made available using a well known method such as random-replacement (where a random IP address is expunged) or least-recently-used (where the least recently used IP address in the table is expunged) to free up a slot for the new IP address.

*Denker*, col. 10, l. 50 through col. 11, l. 24.

The cited passage manifestly does not show or suggest the use of a key as claimed. The invention of claim 3 is directed to using a key for the Initial Sequence number Receiver side.



The cited portion of *Denker* merely teaches randomly replacing entries in the friends table when the friends table becomes full. As shown above, the friends table has nothing to do with establishing a connection, whereas the claimed invention is directed towards establishing a connection in the first place. Thus, the cited portion of *Denker* has nothing to do with the invention of claim 3 and certainly does not show or suggest any of the limitations of claim 3. Furthermore, *Denker* is devoid of disclosure with respect to claim 3. Accordingly, *Denker* does not anticipate claim 3.

Claims 10 and 13 stand or fall with claim 3.

### A.3 Claim 4

The examiner rejects claim 4 as anticipated by *Denker*, further stating that:

As to dependent claim 4, "wherein the step of concatenating said server signature and said category index further includes the step of picking up a category index within said set of predefined connection categories on the basis of the content of said received SYN message" is disclosed in 053 col. 7, lines 47-67.

Office Action of January 25, 2005, p. 4-5 (emphasis in original).

Claim 4 is as follows:

4. The method according to claim 2, wherein the step of concatenating said server signature and said category index further includes the step of:  
picking a category index within said set of connection categories  
on the basis of content of said SYN message.

*Denker* does not anticipate claim 4 because *Denker* does not disclose picking a category index within said set of connection categories on the basis of content of said SYN message, as claimed. The examiner asserts otherwise, citing the following portions of *Denker*:

The SYNACK message of step 2030C also includes an encoded value (represented in FIG. 4 as \$c). For security reasons, the encoded value \$c can be calculated by server 110 as a cryptologic function (or other mathematical function) that depends upon at least the IP address of client 105 and a secret only known to server 110. The encoded value \$c can be a cryptologic function which depends upon one or more additional parameters (in addition to the secret and the IP address of client 105), including: the client's port, the server's IP address, the server's port, and the client's sequence number, among other things. For example, the encoded value \$c can be calculated by server 110 as follows:

$S_c$ =MD5 hash (client's IP address, client's port, server's IP address, server's port, random secret)+client's initial sequence number.(Eq. 1).

Equation 1 states that the encoded value  $S_c$  can be calculated as the MD5 hash function of the client's IP address, the client's port, the server's IP address, the server's port and the random secret known only to server 110 plus the client's initial sequence number (shown as 100 in message 1).

*Denker*, col. 7, ll. 47-67

The cited passage manifestly does not show or suggest picking a category index, as claimed. Nowhere does *Denker* show or suggest picking a category index, as claimed. Thus, *Denker* does not anticipate claim 4.

#### A.4 Claim 5

The examiner rejects claim 5 as anticipated by *Denker*, further stating that:

**As to dependent claim 5, "wherein said updating step includes the step of: updating said PRN generator at a rate not higher than a Maximum Segment Lifetime defined in said transmission control protocol connections" is taught in '053, col. 7, lines 47-61.**

Office Action of January 25, 2005, p. 5 (emphasis in original)

Claim 5 is as follows:

5. The method according to claim 3, wherein said updating step includes the step of:  
updating said PRN generator at a rate not higher than a Maximum Segment Lifetime defined in said transmission control protocol connections.

*Denker* does not anticipate claim 5 because *Denker* shows none of the limitations of claim

5. The examiner asserts otherwise, citing the following passages from *Denker*:

The SYNACK message of step 2030C also includes an encoded value (represented in FIG. 4 as  $S_c$ ). For security reasons, the encoded value  $S_c$  can be calculated by server 110 as a cryptologic function (or other mathematical function) that depends upon at least the IP address of client 105 and a secret only known to server 110. The encoded value  $S_c$  can be a cryptologic function which depends upon one or more additional parameters (in addition to the secret and the IP address of client 105), including: the client's port, the server's IP address, the server's port, and the client's sequence number, among other things. For example, the encoded value  $S_c$  can be calculated by server 110 as follows:

$Sc = \text{MD5 hash (client's IP address, client's port, server's IP address, server's port, random secret)} + \text{client's initial sequence number. (Eq. 1)}$

*Denker*, col. 7, ll. 47-61

The cited passage manifestly does not show or suggest updating a PRN generator at a maximum rate, as claimed. Furthermore, *Denker* is devoid of disclosure regarding the invention of claim 5. Accordingly, *Denker* does not anticipate claim 5.

#### A.5 Claim 7

The examiner rejects claim 7 as anticipated by *Denker*, stating that:

as to independent claim 7, "A method for defeating, in a server unit of an IP network, a SYN flooding attack, said method comprising the steps of:" is disclosed in '053 col. 4, lines 33-54;

"listening for an ACK message sent from a client unit" and "resuming said listening step" is taught in '053 col. 6, lines 59-60;

"upon receiving said ACK message, evaluating a value of an Initial Sequence number Receiver side that includes content comprising embedded connection parameters specified in a previously received SYN message as an authentic computed Initial Sequence number Receiver side; and responsive to evaluating the value of the Initial Sequence Number Receiver side as an authentic computed Initial Sequence number Receiver side, allocating resources for a transmission control protocol connection according to said content; and" is shown in '053 col. 5, line 1-43.

Office Action of January 25, 2005, p. 5 (emphasis in original).

Claim 7 is as follows:

7. A method for defeating, in a server unit of an IP network, a SYN flooding attack, said method comprising the steps of:
  - listening for an ACK message sent from a client unit;
  - upon receiving said ACK message, evaluating a value of an Initial Sequence Number Receiver side that includes content comprising embedded connection parameters specified in a previously received SYN message as an authentic computed Initial Sequence Number Receiver side; and
  - responsive to evaluating the value of the Initial Sequence Number Receiver side as an authentic computed Initial Sequence Number Receiver side, allocating resources for a transmission control protocol connection according to said content; and
  - resuming said listening step.

*Denker* does not anticipate claim 7 because *Denker* does not show the step of: responsive to evaluating the value of the Initial Sequence Number Receiver side as an authentic computed Initial Sequence Number Receiver side, allocating resources for a transmission control protocol connection according to said content, as claimed. The examiner asserts otherwise, citing the following passages from *Denker*:

The server then analyzes the encoded value in the ACK message to determine if it passes the appropriate mathematical (i.e., cryptologic) test. If the encoded value included in the ACK message passes the appropriate mathematical test, then the client is properly complying with the TCP2B protocol, and the server allocates a full Transmission Control Block in memory, and the connection becomes fully established.

In the TCP2E protocol according to an embodiment of the present invention, the server maintains a Friends Table, which is a list of addresses of those devices recently observed to be complying with TCP. The client requests a TCP connection with a server using a SYN message. The server then determines whether the address of the client is on the server's Friends Table.

If the client's address is not on the server's Friends Table, the server calculates an encoded value. The encoded value is calculated as the mathematical function of at least the client's address and a secret (i.e., a random number) known only to the server. The server sends an ACK message to the client including the calculated encoded value as the acknowledgment number. Because the acknowledgment number does not acknowledge any messages previously sent by the client, this ACK message appears to the client as a half-open connection. The client responds by sending a reset message to the server, as required by the standard TCP specifications. If the reset message passes a mathematical test at the server, the server adds the client's address to the Friends Table. Then, in accordance with standard TCP, the client will again attempt to establish a TCP connection with the server by re-issuing the SYN message to the server. This SYN message (or packet) will be welcomed by the server, since the client's address is now in the Friends Table.

According to TCP2E, if the client's address is on the server's Friends Table, the connection request (i.e., received SYN message) is processed according to TCP. A counter associated with each address in the Friends Table can be used to keep track of the number of successful connections established as compared to the total number of connection requests from the client, and allow the server to expunge the client's address from the Friends Table if there are too many unsuccessful connection attempts.


*Denker*, col. 5, ll. 1-43.

The cited passage manifestly does not show or suggest evaluating the Initial Sequence number Receiver side and allocating resources in the manner claimed. *Denker* is devoid of disclosure regarding the claimed limitation. Thus, *Denker* does not anticipate claim 7.

Claims 8 and 9 stand or fall with claim 7.

#### B. SUMMARY

Claims 2, 11, 12, and 14 stand or fall with claim 1. Claims 8 and 9 stand or fall with claim 7. Claims 10 and 13 stand or fall with claim 3. Because *Denker* does not show or suggest all of the limitations of any of the claims, *Denker* does not anticipate any of the claims. Accordingly, Applicants respectfully request that the Board overturn the rejections and order that the claims be allowed.

  
\_\_\_\_\_  
Theodore D. Fay, III  
Reg. No. 48,504  
YEE & ASSOCIATES, P.C.  
PO Box 802333  
Dallas, TX 75380  
(972) 385-8777  
ATTORNEY FOR APPLICANTS

**CLAIMS APPENDIX**

The text of the claims involved in the appeal are:

1. (Previously Presented) A method for defeating, in a server unit of an Internet Protocol network, a SYN flooding attack, said server unit running Transmission Control Protocol to allow the establishment of one or more transmission control protocol connections with one or more client units, said method comprising the steps of:
  - upon having activated the transmission control protocol in said server unit,
  - listening for the receipt of a SYN message sent from a client unit;
  - upon receiving said SYN message,
  - computing an Initial Sequence number Receiver side, wherein said Initial Sequence number Receiver side is embedded with connection parameters specified in the SYN message;
  - responding to said client unit with a SYN-ACK message including said Initial Sequence number Receiver side;
  - resuming to said listening step; and
  - responsive to receiving an ACK message, determining whether to establish a transmission control block for the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message.

2. (Previously Presented) The method according to claim 1 wherein the step of computing said Initial Sequence number Receiver side further includes the steps of:

concatenating a randomly generated key with an identification of one of said transmission

control protocol connections, said identification including:

a client socket and a server socket;

a server signature calculated by hashing said concatenation; and

a concatenation of said server signature and a category index referring to a set of

predefined transmission control protocol connection categories.

3. (Previously Presented) The method according to claim 2, wherein said computing step further comprises the steps of:

updating, in said server unit, a pseudo-random number (PRN) generator;

holding a current key;

remembering a former key; and

using said current key as said randomly generated key for said Initial Sequence number Receiver side.

4. (Previously Presented) The method according to claim 2, wherein the step of concatenating said server signature and said category index further includes the step of:

picking a category index within said set of connection categories on the basis of content of said SYN message.

5. (Previously Presented) The method according to claim 3, wherein said updating step includes the step of:

updating said PRN generator at a rate not higher than a Maximum Segment Lifetime defined in said transmission control protocol connections.

6. (Canceled)

7. (Previously Presented) A method for defeating, in a server unit of an IP network, a SYN flooding attack, said method comprising the steps of:

listening for an ACK message sent from a client unit;

upon receiving said ACK message, evaluating a value of an Initial Sequence Number

Receiver side that includes content comprising embedded connection parameters specified in a previously received SYN message as an authentic computed Initial Sequence Number Receiver side; and

responsive to evaluating the value of the Initial Sequence Number Receiver side as an authentic computed Initial Sequence Number Receiver side, allocating resources for a transmission control protocol connection according to said content; and resuming said listening step.

8. (Previously Presented) The method of claim 7, further including:

interpreting a category index extracted from said value of the Initial Sequence Number Receiver side .



9. (Previously Presented) The method according to claim 8, wherein the allocating step includes the step of:

selecting a predefined set of parameters, for said transmission control protocol connection, on the basis of the category index.

10. (Previously Presented) The method according to claim 7, wherein the step of evaluating said Initial Sequence Number Receiver side includes, upon receiving said ACK message, the steps of:

having, firstly, selected a current key:

getting said current key;

concatenating said current key with an identification of said transmission control protocol connection, said identification including:

a client socket and a server socket;

hashing said concatenation of the current key and the identification, thus obtaining a re-computed server signature;

extracting an acknowledgment field from said ACK message;

decrementing content of said acknowledgement field;

extracting a server signature from the decremented content; and

comparing said re-computed server signature and said extracted server signature.

11. (Previously Presented) A computer program product for defeating, in a server unit of an Internet Protocol network, a SYN flooding attack, said server unit running Transmission Control Protocol to allow the establishment of one or more transmission control protocol connections

with one or more client units, said computer program product having computer readable program code comprising:

computer readable program code, responsive to having activated the transmission control protocol in said server unit, for listening for the receipt of a SYN message sent from a client unit;

computer readable program code for computing an Initial Sequence number Receiver side responsive to receiving said SYN message, wherein said Initial Sequence number Receiver side includes embedded connection parameters ;

computer readable program code for responding to said client unit with a SYN-ACK message including said Initial Sequence number Receiver side;

computer readable program code for resuming said listening step; and

computer readable program code for, responsive to receiving an ACK message, determining whether to establish a transmission control block for the client unit by evaluating an incremented value of the Initial Sequence number Receiver side included in the ACK message.

12. (Previously Presented) The computer program product according to claim 11, wherein the computer readable program code for computing said Initial Sequence number Receiver side further includes:

computer readable program code for calculating a concatenation of a randomly generated key with an identification of one of said one or more transmission control protocol connections, said identification including:

a client socket and a server socket;

a server signature calculated by hashing said concatenation; and  
a concatenation of said server signature and a category index referring to a set of  
predefined transmission control protocol connection categories.

13. (Previously Presented) The computer program product according to claim 11 or 12  
wherein said computing step further comprises the steps of:

computer readable program code means for updating, in said server unit, a pseudo-  
random number (PRN) generator;  
computer readable program code for holding a current key;  
computer readable program code for remembering a former key; and  
computer readable program code for using said current key as the former key for  
evaluating said Initial Sequence number Receiver side.

14. (Previously Presented) A system for implementing a shield for defeating TCP SYN  
flooding attacks, said system comprising:

an Internet Protocol network;  
a server unit running Transmission Control Protocol to allow the establishment of one or  
more transmission control protocol connections; and  
one or more client units; wherein, once said Transmission Control Protocol is activated in  
said server unit, said server unit listens for the receipt of a SYN message from one  
or more of said client units, and whereupon receiving said SYN message from a  
client unit, said server unit computes an Initial Sequence number Receiver side  
having connection parameters embedded therein, responds to said client unit with

a SYN-ACK message including said Initial Sequence number Receiver side and resumes listening for further SYN messages, and wherein said server unit, responsive to receiving an ACK message, determines whether to establish a transmission control block for the client unit by evaluating a value comprising an increment of the Initial Sequence number Receiver side included in the ACK message.

**EVIDENCE APPENDIX**

There is no additional evidence to be presented.

**RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.